Carnegie Mellon University
Software Engineering Institute

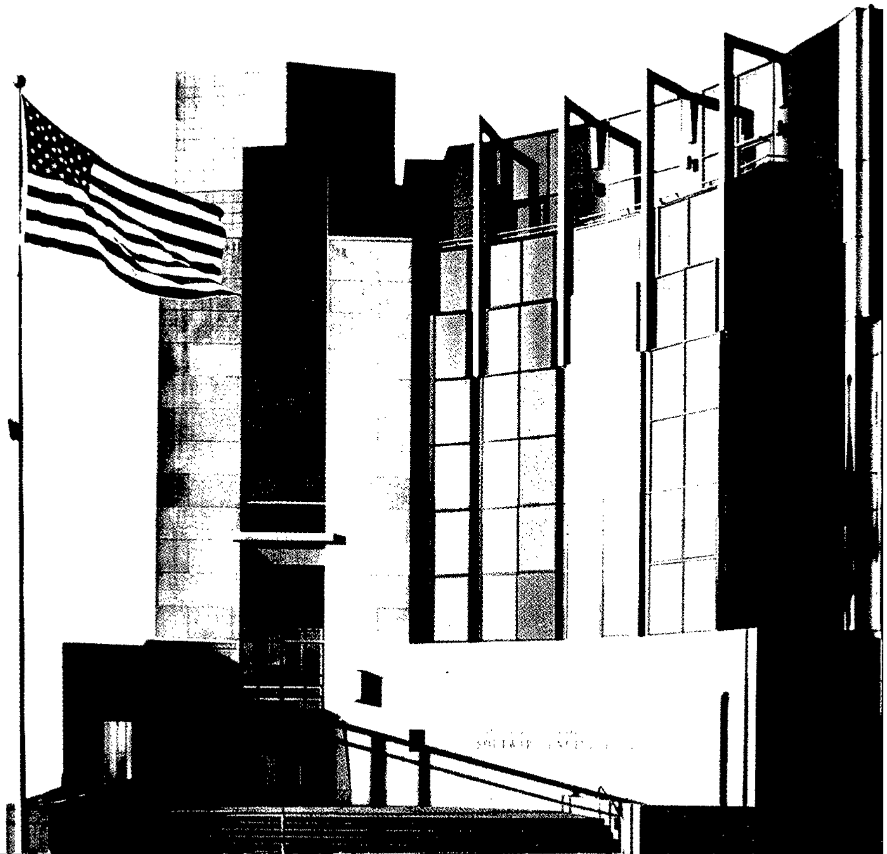
Survivable Network Systems: An Emerging Discipline

R. J. Ellison
D. A. Fisher
R. C. Linger
H. F. Lipson
T. Longstaff
N. R. Mead

November 1997

TR

TECHNICAL REPORT
CMU/SEI-97-TR-013
ESC-TR-97-013



Technical Report
CMU/SEI-97-TR-013
ESC-TR-97-013
November 1997

Survivable Network Systems: An Emerging Discipline

R. J. Ellison

D. A. Fisher

R. C. Linger

H. F. Lipson

T. Longstaff

N. R. Mead

Survivable Network Technology Team
CERT®

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

Jay Alonis, Lt Col, USAF

SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This document is available through SAIC/ASSET: 1350 Earl L. Core Road; PO Box 3305; Morgantown, West Virginia 26505 / Phone: (304) 284-9000 / FAX: (304) 284-9001 / World Wide Web: http://www.asset.com/SEI.html
/ e-mail: sei@asset.com.

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / Attn: BRR / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218. Phone: (703) 767-8274 or toll-free in the U.S. 1-800 225-3842).

Table of Contents

List of Tables

List of Figures

Survivable Network Systems:
An Emerging Discipline

Abstract: Society is growing increasingly dependent upon large-scale, highly distributed systems that operate in unbounded network environments. Unbounded networks, such as the Internet, have no central administrative control and no unified security policy. Furthermore, the number and nature of the nodes connected to such networks cannot be fully known. Despite the best efforts of security practitioners, no amount of system hardening can assure that a system that is connected to an unbounded network will be invulnerable to attack. The discipline of survivability can help ensure that such systems can deliver essential services and maintain essential properties such as integrity, confidentiality, and performance, despite the presence of intrusions. Unlike the traditional security measures that require central control or administration, survivability is intended to address unbounded network environments. This report describes the survivability approach to helping assure that a system that must operate in an unbounded network is robust in the presence of attack and will survive attacks that result in successful intrusions. Included are discussions of survivability as an integrated engineering framework, the current state of survivability practice, the specification of survivability requirements, strategies for achieving survivability, and techniques and processes for analyzing survivability.

1. Survivability in Network Systems

Contemporary large-scale networked systems that are highly distributed improve the efficiency and effectiveness of organizations by permitting whole new levels of organizational integration. However, such integration is accompanied by elevated risks of intrusion and compromise. These risks can be mitigated by incorporating survivability capabilities into an organization's systems. As an emerging discipline, survivability builds on related fields of study (e.g., security, fault tolerance, safety, reliability, reuse, performance, verification, and testing) and introduces new concepts and principles. Survivability focuses on preserving essential services in unbounded environments, even when systems in such environments are penetrated and compromised [Anderson 97].

1.1 The New Network Paradigm: Organizational Integration

From their modest beginnings some 20 years ago, computer networks have become a critical element of modern society. These networks not only have global reach, they also have impact on virtually every aspect of human endeavor. Network systems are principal enabling agents in business, industry, government, and defense. Major economic sectors, including defense, energy, transportation, telecommunications, manufacturing, financial services, health care, and education, all depend on a vast array of networks operating on local, national, and global scales. This pervasive societal dependency on networks magnifies the consequences of intrusions, accidents, and failures, and amplifies the critical importance of ensuring network survivability.

As organizations seek to improve efficiency and competitiveness, a new network paradigm is emerging. Networks are being used to achieve radical new levels of organizational integration. This integration obliterates traditional organizational boundaries and transforms local operations into components of comprehensive, network-resident business processes. For example, commercial organizations are integrating operations with business units, suppliers, and customers through large-scale networks that enhance communication and services. These networks combine previously fragmented operations into coherent processes open to many organizational participants. This new paradigm represents a shift from bounded networks with central control to unbounded networks. Unbounded networks are characterized by distributed administrative control without central authority, limited visibility beyond the boundaries of local administration, and lack of complete information about the network. At the same time, organizational dependencies on networks are increasing and risks and consequences of intrusions and compromises are amplified.

1.2 The Definition of Survivability

We define survivability as the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents. We use the term *system* in the broadest possible sense, including networks and large-scale systems of systems.

The term *mission* refers to a set of very high-level (i.e., abstract) requirements or goals. *Mission*s are not limited to military settings since any successful organization or project must have a vision of its objectives whether expressed implicitly or as a formal mission statement. Judgements as to whether or not a mission has been successfully fulfilled are typically made in the context of external conditions that may affect the achievement of that mission. For example, assume that a financial system shuts down for 12 hours during a period of widespread power outages caused by a hurricane. If the system preserves the integrity and confidentially of its data and resumes its essential services after the period of environmental stress is over, the system can reasonably be judged to have fulfilled its mission. However, if the same system shuts down unexpectedly for 12 hours under normal conditions (or under relatively minor environmental stress) and deprives its users of essential financial services, the system can reasonably be judged to have failed its mission, even if data integrity and confidentiality are preserved.

Timeliness is a critical factor that is typically included in (or implied by) the very high-level requirements that define a mission. However, timeliness is such an important factor that we included it explicitly in the definition of survivability.

The terms *attack*, *failure*, and *accident* are meant to include all potentially damaging events; but these terms do not partition these events into mutually exclusive or even distinguishable sets. It is often difficult to determine if a particular detrimental event is the result of a malicious attack, a failure of a component, or an accident. Even if the cause is eventually determined, the critical immediate response cannot depend on such speculative future knowledge.

Attacks are potentially damaging events orchestrated by an intelligent adversary. Attacks include intrusions, probes, and denial of service. Moreover, the threat of an attack may have as severe an impact on a system as an actual occurrence. A system that assumes a defensive position because of the threat of an attack may reduce its functionality and divert additional resources to monitoring the environment and protecting system assets.

We include failures and accidents as part of survivability. *Failures* are potentially damaging events caused by deficiencies in the system or in an external element on which the system depends. Failures may be due to software design errors, hardware degradation, human errors, or corrupted data. *Accidents* describe the broad range of randomly occurring and potentially damaging events such as natural disasters. We tend to think of accidents as externally generated events (i.e., outside the system) and failures as internally generated events.

With respect to system survivability, a distinction between a failure and an accident is less important than the impact of the event. Nor is it often possible to distinguish between intelligently orchestrated attacks and unintentional or randomly occurring detrimental events. Our approach concentrates on the effect of a potentially damaging event. Typically, for a system to survive, it must react to (and recover from) a damaging effect (e.g., the integrity of a database is compromised) long before the underlying cause is identified. In fact, the reaction and recovery must be successful whether or not the cause is ever determined.

Our primary focus in this report is to help systems survive the acts of intelligent adversaries. This bias is based on the nature of the organization to which the authors belong. Our Survivable Network Technology Team is an outgrowth of the CERT® Coordination Center, which has been helping users respond to and recover from computer security incidents since 1988.

Finally, it is important to recognize that it is the mission fulfillment that must survive, not any particular subsystem or system component. Central to the notion of survivability is the capability of a system to fulfill its mission, even if significant portions of the system are damaged or destroyed. We will sometimes use the term *survivable system* as a less than perfectly precise shorthand for a system with the capability to fulfill a specified mission in the face of attacks, failures, or accidents. Again, it is the mission, not a particular portion of the system, that must survive.

1.3 The Domain of Survivability: Unbounded Networks

The success of a survivable system depends on the computing environment in which the survivable system operates. The trend in networked computing environments is towards largely unbounded network infrastructures. A bounded system is one in which all of the system's parts are controlled by a unified administration and can be completely characterized and controlled. At least in theory, the behavior of a bounded system can be understood and all of its various parts identified. In an unbounded system there is no unified administrative control over its parts. We use the term *administrative control* in the strictest sense, which includes the power to impose and enforce sanctions and not simply to recommend an appropriate security policy. In an unbounded system, each participant has an incomplete view of the whole, must depend on and trust information supplied by its neighbors, and cannot exercise control outside its local domain.

® CERT is registered in the U.S. Patent and Trademark Office.

An unbounded system can be composed of bounded and unbounded systems connected together in a network. Figure 1 illustrates an unbounded domain consisting of a collection of bounded systems in which each bounded system is under separate administrative control. Although the security policy of an individual bounded system cannot be fully enforced outside of the boundaries of its administrative control, the policy can be used as a yardstick to evaluate the security state of that bounded system. Of course, the security policy can be advertised outside of the bounded system; but administrators are severely limited in their ability to compel or persuade outside individuals or entities to follow it. This limitation is particularly true when an unbounded domain spans jurisdictional boundaries, making legal sanctions difficult or impossible to impose.

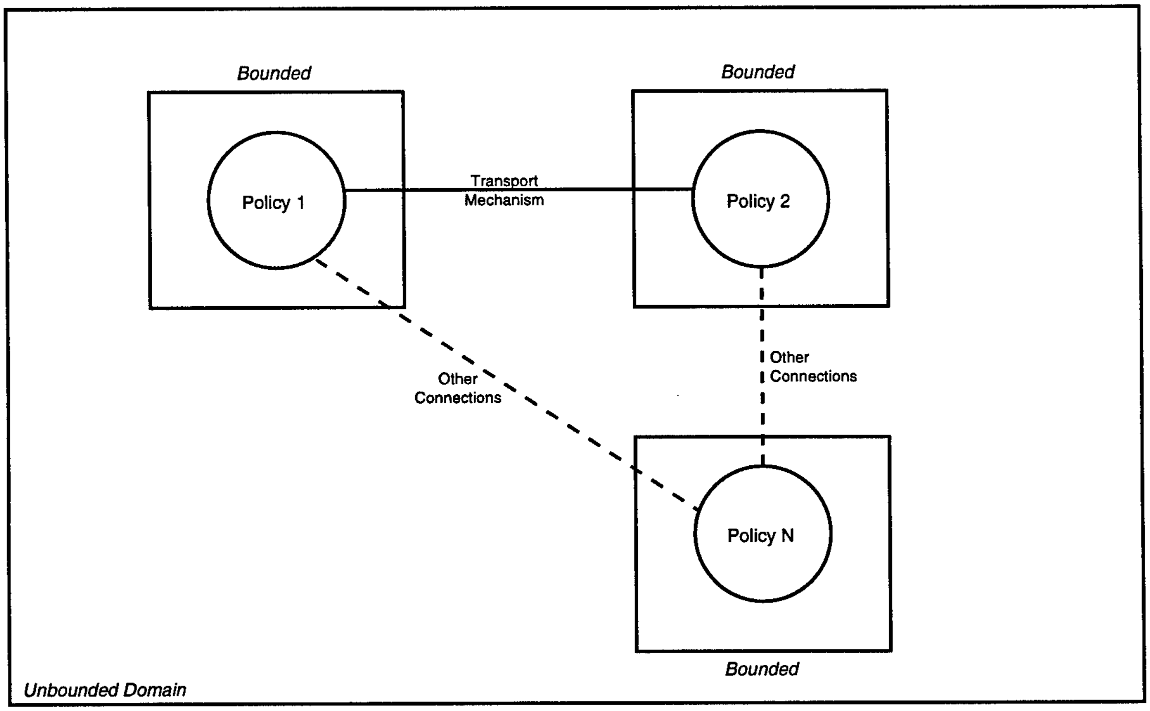


Figure 1: An Unbounded Domain Viewed as a Collection of Bounded Systems

When an application or software-intensive system is exposed to an environment consisting of multiple, unpredictable administrative domains with no measurable bounds, the system has an unbounded environment. An unbounded environment exhibits the following properties:

- multiple administrative domains with no central authority
- an absence of global visibility (i.e., the number and nature of the nodes in the network cannot be fully known)
- interoperability between administrative domains determined by convention
- widely distributed and interoperable systems
- users and attackers can be peers in the environment
- cannot be partitioned into a finite number of bounded environments

The Internet is an example of an unbounded environment with many client-server network applications. A public Web server and its clients may exist within many different administrative domains on the Internet; yet there exists no central authority that requires all clients to be configured in a way expected by the Web server. In particular, a Web server can never rely on a set of client plug-ins to be present or absent for any function that the server may want to provide.

For a Web server providing a financial transaction (e.g., for a Web-based purchase), the Web server may require that the user install a plug-in on the client to support a secure transaction. However, due to the unbounded nature of the environment, previously installed plug-ins from a competitor may be present on the client that may corrupt, subvert, or damage the Web server during the transaction. For the Web server to be survivable, there must be built-in protection from malicious client interactions and these protections must make no assumptions about the configuration or features of the remote client.

In this example, the Web server and its clients make up the system. The multiple administrative domains are the variety of site domains on the Internet. Many of these domains have legitimate users. Other sites are used for intrusions in an anonymous setting. These latter sites cannot be distinguished by their administrative domain, but only by client behavior. The interoperability between the server and its clients is defined by http (hypertext transfer protocol), a convention agreed upon between the server and clients. The system, comprised of Web servers and clients, is widely distributed both geographically and logically throughout the Internet. Legitimate users and attackers are peers in the environment and there is no method to isolate legitimate users from the attackers. In other words, there is no way to bound the environment to legitimate users using only a common administrative policy.

Unbounded systems are a significant component of today's computing environment and will play an even a larger role in the future. The Internet — a non-hierarchical network of systems, each under local administrative control only — is a primary example of an unbounded system. While conventions exist that allow the parts of the Internet to work together, there is no global administrative control to assure that these parts behave according to these conventions. Therefore, security problems abound. Unfortunately, the security problems associated with unbounded systems are typically underestimated.

1.4 Characteristics of Survivable Systems

A key characteristic of survivable systems is their capability to deliver essential services in the face of attack, failure, or accident.

Central to the delivery of essential services is the capability of a system to maintain essential properties (i.e., specified levels of integrity, confidentiality, performance, and other quality attributes) in the presence of attack, failure, or accident. Thus, it is important to define minimum levels of quality attributes that must be associated with essential services. For example, a launch of a missile by a defensive system is no longer effective if the system performance is slowed to the point that the target is out of range before the system can launch.

These quality attributes are so important that definitions of survivability are often expressed in terms of maintaining a balance among multiple quality attributes such as performance, security, reliability, availability, fault-tolerance, modifiability, and affordability. The Attribute Tradeoff Analysis project at the Software Engineering Institute is using this attribute-balancing (i.e., tradeoff) view of survivability to evaluate and synthesize survivable systems [Kazman 97]. Quality attributes represent broad categories of related requirements, so a quality attribute may contain other quality attributes. For example, the security attribute traditionally includes the three attributes: availability, integrity, and confidentiality.

The capability to deliver essential services (and maintain the associated essential properties) must be sustained even if a significant portion of the system is incapacitated. Furthermore, this capability should not be dependent upon the survival of a specific information resource, computation, or communication link. In a military setting, *essential services* might be those required to maintain an overwhelming technical superiority, and *essential properties* may include integrity, confidentiality, and a level of performance sufficient to deliver results in less than one decision cycle of the enemy. In the public sector, a survivable financial system is one that maintains the integrity, confidentiality, and availability of essential information and financial services, even if particular nodes or communication links are incapacitated through intrusion or accident, and that recovers compromised information and services in a timely manner. The financial system's survivability might be judged by using a composite measure of the disruption of stock trades or bank transactions (i.e., a measure of the disruption of essential services).

Key to the concept of survivability, then, is identifying the essential services (and the essential properties that support them) within an operational system. *Essential services* are defined as the functions of the system that must be maintained when the environment is hostile or failures or accidents are detected that threaten the system. There are typically many services that can be temporarily suspended when a system is dealing with an attack or other extraordinary environmental condition. Such a suspension can help isolate areas affected by an intrusion and free system resources to deal with its effects. The overall function of a system should adapt to preserve essential services.

We have linked the capability of a survivable system to fulfill its mission in a timely manner to its ability to deliver essential services in the presence of attack, accident, or failure. Ultimately, mission fulfillment must survive, not any portion or component of the system. If an *essential service* is lost, it can be replaced by another service that supports mission fulfillment in a different but equivalent way. However, we still believe that the identification and protection of essential services is an important part of a practical approach to building and analyzing survivable systems. As a result, we define *essential services* to include alternate sets of essential services (perhaps mutually exclusive) that need not be simultaneously available. For example, a set of essential services to support power delivery may include both the distribution of electricity and the operation of a natural gas pipeline.

To maintain their capabilities to deliver essential services, survivable systems must exhibit the four key properties illustrated in Table 1:

Key Property	Description	Example
Resistance to attacks	strategies for repelling attacks	user authentication stochastic diversity of programs
Recognition of attacks and the extent of damage	strategies for detecting attacks (including intrusions) and understanding the current state of the system, including evaluating the extent of damage	recognition of intrusion usage patterns internal integrity checking
Recovery of full and essential services after attack	strategies for restoring compromised information or functionality, limiting the extent of damage, maintaining or, if necessary, restoring essential services within the time constraints of the mission, restoring full service as conditions permit	replication and reinitialization of data
Adaptation and evolution to reduce effectiveness of future attacks	strategies for improving system survivability based on knowledge gained from intrusions	incorporation of new patterns for intrusion recognition

Table 1: The Key Properties of Survivable Systems

1.5 Survivability as an Integrated Engineering Framework

As a broadly-based engineering paradigm, survivability is a natural framework for integrating established and emerging software engineering disciplines in the service of a common goal. These established areas of software engineering, which are related to survivability, include security, fault tolerance, safety, reliability, reuse, performance, verification, and testing. Research in survivability encompasses a wide variety of research methods, including the investigation of

- analogs to the immunological functioning of an individual organism
- sociological analogs to public health efforts at the community level

1.5.1 Survivability and Security

The discipline of computer security has made valuable contributions to the protection and integrity of information systems over the past three decades. However, *computer security* has traditionally been used as a binary term that suggests that at any moment in time a system is either safe or compromised. We believe that this use of *computer security* engenders viewpoints that largely ignore the aspects of recovery from the compromise of a system and aspects of maintaining services during and after an intrusion. Such an approach is inadequate to support necessary improvements in the state of the practice of protecting computer systems from attack. In contrast, the term *survivable systems* refers to systems whose components collectively accomplish their mission even under attack and despite active intrusions that effectively damage a significant portion of the system.

Robustness under attack is at least as important as hardness or resistance to attack. Hardness contributes to survivability, but robustness under attack (and, in particular, recoverability) is the essential characteristic that distinguishes survivability from traditional computer security. At the same time, survivability can benefit from computer security research and practice, and survivability can provide a framework for integrating security with other disciplines that can contribute to system survivability.

1.5.2 Survivability and Fault Tolerance

Survivability requires robustness under conditions of intrusion, failure, or accident. The concept of survivability includes fault tolerance, but is not equivalent to it. Fault tolerance relates to the statistical probability of an accidental fault or combination of faults, not to malicious attack. For example, an analysis of a system may determine that the simultaneous occurrence of the three statistically independent faults (f1, f2, and f3) will cause the system to fail. The probability of the three independent faults occurring simultaneously by accident may be extremely small, but an intelligent adversary with knowledge of the system's internals can orchestrate the simultaneous occurrence of these three faults and bring down the system. A fault-tolerant system most likely does not address the possibility of the three faults occurring simultaneously, if the probability of occurrence is below a threshold of concern. A survivable system requires a contingency plan to deal with such a possibility.

Redundancy is another factor that can contribute to the survivability of systems. However, redundancy alone is insufficient since multiple identical backup systems share identical vulnerabilities. A survivable system requires each backup system to offer equivalent functionality, but significant variance in implementation. This variance thwarts attempts to compromise the primary system and all backup systems with a single attack strategy.

1.6 The Current State of Practice in Survivable Systems

Much of today's research and practice in computer-systems survivability takes a perilously narrow, security-based view of defense against computer intrusions. This narrow view is dangerously incomplete because it focuses almost exclusively on hardening a system (e.g., using firewall technology or an orange book approach to host protection) to prevent a break-in or other malicious attack. This view does little about how to detect an intrusion or what to do once an intrusion has occurred or is under way. This view is also accompanied by evaluation techniques that limit their focus to the relative hardness of a system, as opposed to a system's robustness under attack and ability to recover compromised capabilities.

The architecture of secure bounded systems is built upon the existence of a security policy and its enforcement, which is imposed by the exercise of administrative control. In contrast, an unbounded system has no administrative control with which to impose global-security policy. For instance, on the Internet today the backbone architecture exists independent of security policy considerations because there is no global administrative control.

Affordability is always a significant factor in the design, implementation, and maintenance of systems that support the national infrastructure (e.g., the power grid, the public switched communications networks, and the financial networks) and our national defense. In fact, the trend toward increased sharing of common infrastructure components in the interest of economy virtually ensures that the civilian networked information infrastructure, and its vulnerabilities will always be an inseparable part of our national defense.

Practical, affordable systems are almost never 100% customized, but rather are constructed from commonly available off-the-shelf components with internal structures that are well known. The trend toward developing systems through integration and reuse instead of customized design and coding efforts is a cornerstone of modern software engineering. Unfortunately, the intellectual complexity associated with software design, coding, and testing virtually ensures that exploitable bugs can and will be discovered in commercial and public domain products with internal structures that are available for analysis. When these products are incorporated as components of larger systems, those systems become vulnerable to attack strategies based on the exploitable bugs. Popular commercial and public-domain components offer attackers a ubiquitous set of targets with well-known and typically unvarying internal structures. This lack of variability among components translates into a lack of variability among systems. These systems potentially allow a single attack strategy to have a wide-ranging and devastating impact.

The natural escalation of offensive threats versus defensive countermeasures has demonstrated time and again that no practical systems can be built that are invulnerable to attack. Despite best efforts, there can be no assurance that systems will not be breached. Thus, the traditional view of information systems security must be expanded to encompass the specification and design of system behavior that helps the system survive in spite of active intrusions. Only then can systems be created that are robust in the presence of attack and are able to survive attacks that cannot be completely repelled.

In short, the nature of contemporary system development dictates that even hardened systems can and will be broken. Therefore, survivability must be designed into systems to help avoid the potentially devastating effects of system compromise and failure due to intrusion.

1.6.1 Incident Handling Has Enhanced Survivability

Although applying the term *survivability* to computer systems is relatively new, the practice of survivability is not. Much of the survivability practice to date has been in the realm of incident response (IR) teams. In fact, the CERT Coordination Center (CERT/CC) has, throughout its history, enhanced system survivability in the Internet community. The CERT/CC provides incident response services (helping organizations respond to and recover from incidents) and publishes and distributes vulnerability advisories (akin to public health notices). Traditionally, the CERT/CC has been concerned about survivability and has been successful in helping sites with risk mitigation and recovery.

The experience of the CERT Coordination Center has shown that how organizations respond to and recover from computer intrusions is at least as important as the steps they take to prevent them. We believe that widespread availability and use of survivable systems by the Internet community and throughout the Internet infrastructure will provide the best hope for the dramatic improvements necessary to transform the Internet into a survivable, networked *information system of systems*. Survivable systems will help make the Internet a viable medium for the conduct of commerce, defense, and government.

This medium will also enable the support of major elements of the national infrastructure (e.g., power grid, public switched network, and air traffic control).

1.6.2 Firewalls Embody the Current State of Practice

Currently, little of the basic technology in security engineering and system integration applies to unbounded systems. Instead, current practice assumes that the capability exists to identify, define, and characterize the extent of administrative control over a system, all access points to that system, and all signals that may appear at those access points. In unbounded systems, such as the current Internet and the future National Information Infrastructure, these boundary conditions cannot be fully determined.

The current state of practice in survivability and security evaluation tends to treat systems and their environments as static and unchanging. However, the survivability and security of systems in fact degrades over time as changes occur in their structures, configurations, and environments, and as knowledge of their vulnerabilities spreads throughout the intruder community.

On the Internet today, the cornerstone of security is the notion of a firewall, a logically bounded system within a physically unbounded one. We assert that *bounded-system thinking* within unbounded domains leads to security designs and architectures that are fundamentally flawed from a survivability perspective. One notable example is the use of a firewall as the basic security component of the Internet. This approach is severely limited and can be readily circumvented by exploiting the fundamental differences between bounded and unbounded systems. Traditional firewalls are the state of the art for security architectures, but not for survivable systems, because they are passive, filter-only devices. The addition of active components, such as detection and a dynamic-response capability, will allow firewalls to play a role in survivable systems; but current firewalls do not have these capabilities.

2. Defining Requirements for Survivable Systems

Survivability requirements can vary substantially depending on system scope, criticality, and the consequences of failure and interruption of service. Categories of requirements definitions for survivable systems include function, use, development, operation, and evolution. In this section, we present what survivability requirements are, how these requirements can be expressed, and their impact on system survivability.

The new paradigm for system requirements definition and design is characterized by distributed services, distributed logic, distributed code (including executable content), distributed hardware, a shared communications and routing infrastructure, diminished trust, and a lack of unified administrative control. Assuring the survivability of mission-critical systems developed under this new paradigm is a formidable high-stakes effort for software engineering research. This effort requires that traditional computer security measures be augmented by new and comprehensive system survivability strategies.

2.1 Expressing Survivability Requirements

The definition and analysis of survivability requirements is a critical first step in achieving system survivability [Linger 97]. Figure 2 depicts an iterative model for defining these requirements. Survivability must address not only requirements for software functionality, but also requirements for software use, development, operation, and evolution. Thus, five types of requirements definitions are relevant to survivable systems in the model. These requirements are discussed in detail in the following subsections.

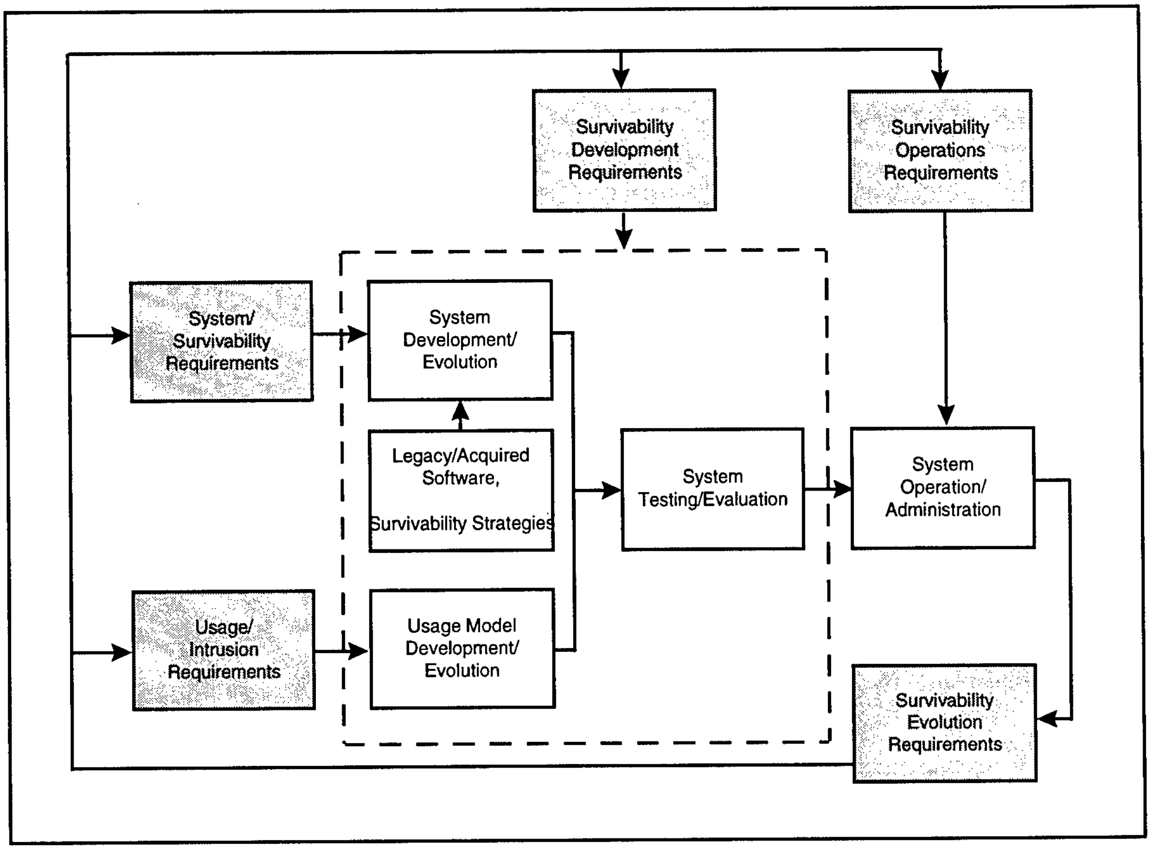

Survivability Development Requirements
Survivability Operations Requirements
System/ Survivability Requirements
System Development/ Evolution
Legacy/Acquired Software,
Survivability Strategies
System Testing/Evaluation
System Operation/ Administration
Usage/ Intrusion Requirements
Usage Model Development/ Evolution
Survivability Evolution Requirements

Figure 2: Requirements Definition for Survivable Systems

System/Survivability Requirements: The term *system requirements* refers to traditional user functions that a system must provide. For example, a network management system must provide functions to enable users to monitor network operations, adjust performance parameters, etc. System requirements also include non-functional aspects of a system, such as timing, performance, and reliability. The term *survivability requirements* refers to the capabilities of a system to deliver essential services in the presence of intrusions and compromises and to recover full services.

Figure 3 depicts the integration of survivability requirements with system requirements at node and network levels.

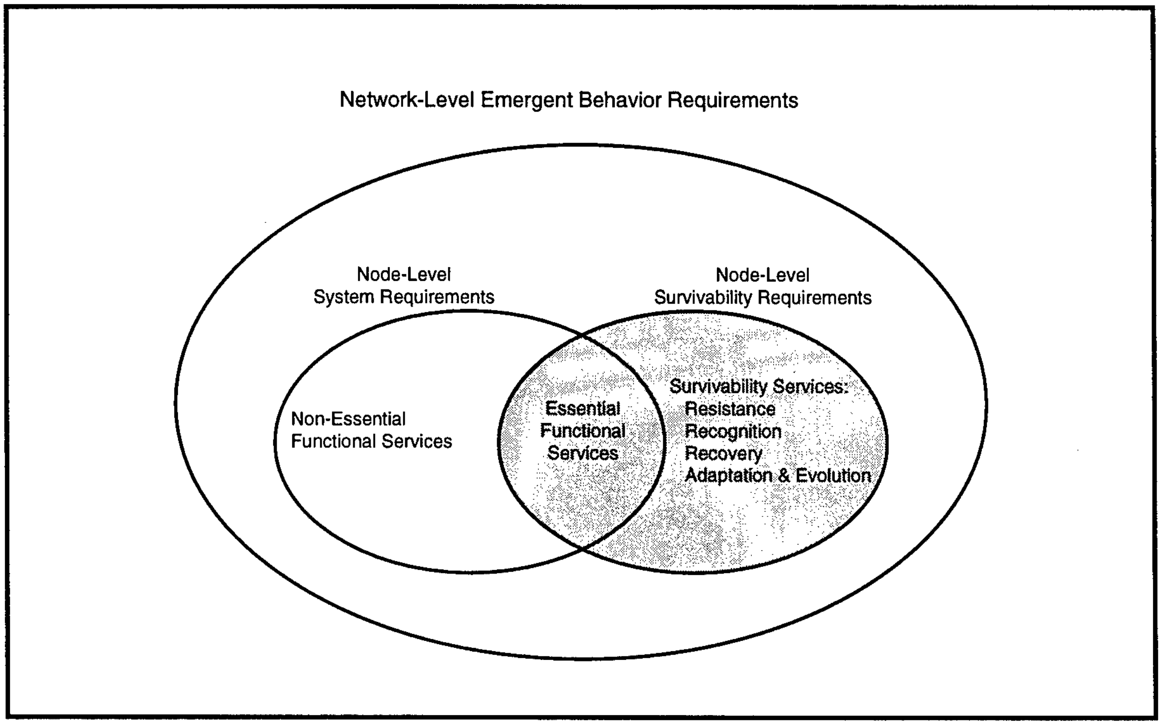

Network-Level Emergent Behavior Requirements
Node-Level System Requirements
Node-Level Survivability Requirements
Non-Essential Functional Services
Essential Functional Services
Survivability Services:
Resistance
Recognition
Recovery
Adaptation & Evolution

Figure 3: Integrating Survivability Requirements with System Requirements

Survivability requires that system requirements be organized into essential services and non-essential services. Essential services must be maintained even during successful intrusions; non-essential services are recovered after intrusions have been handled. Essential services may be stratified into any number of levels, each embodying fewer and more vital services as the severity and duration of intrusion increases. Thus, definitions of requirements for essential services must be augmented with appropriate survivability requirements.

As shown in Figure 2, survivable systems may also include legacy and acquired COTS components that were not developed with survivability as an explicit objective. Such components may provide both essential and non-essential services and may require functional requirements for isolation and control through wrappers and filters to permit their safe use in a survivable system environment.

Figure 3 shows that survivability itself imposes new types of requirements on systems. These new requirements include the *resistance* to, *recognition* of and *recovery* from intrusions and compromises, and *adaptation and evolution* to diminish the effectiveness of future intrusion attempts. These survivability requirements are supported by a variety of existing and emerging *survivability strategies*, as noted in Figure 2 and discussed in more detail below.

Finally, Figure 3 depicts *emergent behavior requirements* at the network level. These requirements are characterized as *emergent* because they are not associated with particular nodes, but rather emerge from the collective behavior of node services in communicating across the network. These requirements deal with the survivability of overall network capabilities (e.g., capabilities to route messages between critical sets of nodes regardless of how intrusions may damage or compromise network topology).

We envision survivable systems that are capable of adapting their behavior, function, and resource allocation in response to intrusions. For example, when necessary, functions and resources devoted to non-essential services could be reallocated to the delivery of essential services and to intrusion resistance, recognition, and recovery. Requirements for such systems must also specify how the system should adapt and reconfigure itself in response to intrusions.

Systems can exhibit large variations in survivability requirements. Small local networks may require few or no essential services and recovery times measured in hours. Conversely, large-scale networks of networks may require a core set of essential services, automated intrusion detection, and recovery times measured in minutes. Embedded command and control systems may require essential services to be maintained in real time and recovery times measured in milliseconds.

The attainment and maintenance of survivability consume resources in system development, operation, and evolution. The resources allocated to a system's survivability should be based on the costs and risks to an organization associated with the loss of essential services.

Use/Intrusion Requirements: Survivable-system testing must demonstrate the correct performance of essential and non-essential system services as well as the survivability of essential services under intrusion. Because system performance in testing (and operation) depends totally on the system's use, an effective approach to survivable-system testing is based on system-use scenarios derived from system-use models [Mills 92, Trammell 95].

System-use models are developed from use requirements that specify use environments and scenarios of system use. Use requirements for essential and non-essential services must be defined in parallel with system and survivability requirements. Furthermore, intruders and legitimate users must be considered equally. Intrusion requirements that specify intrusion-use environments and scenarios of intrusion use must be defined as well. In this approach, intrusion use and legitimate use of system services are modeled together.

Figure 4 depicts the relationship between legitimate and intrusion use. Intruders may engage in scenarios beyond legitimate scenarios, but may also employ legitimate use for purposes of intrusion if they gain the necessary privileges.

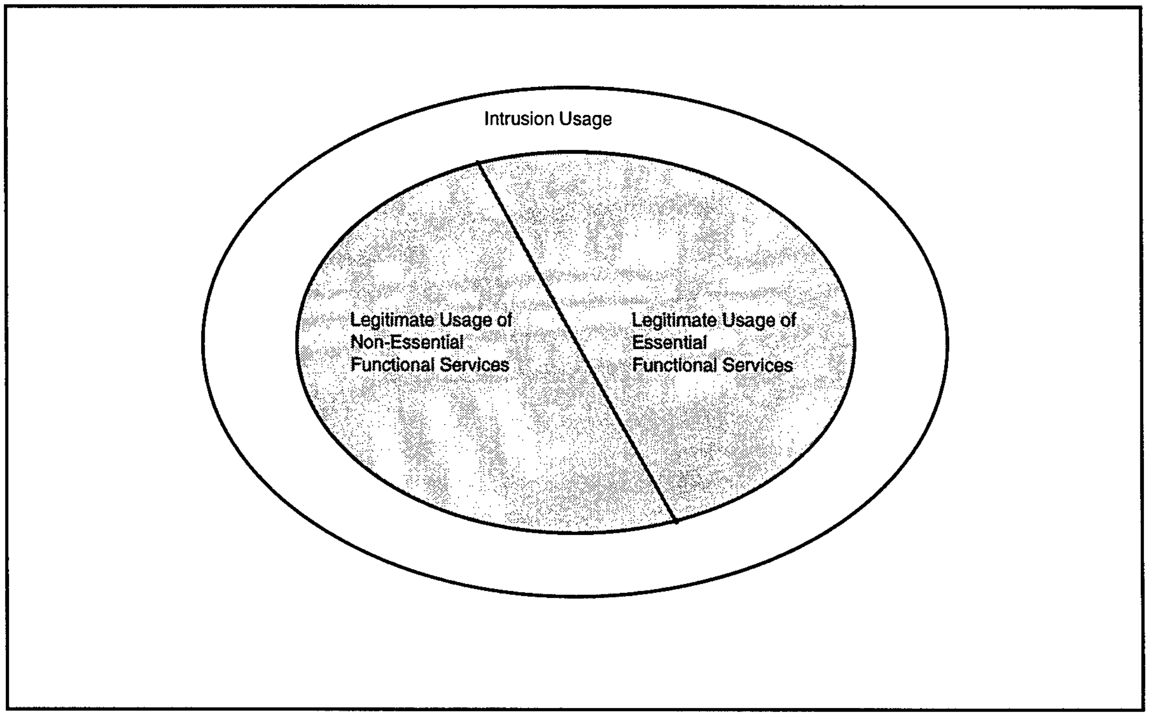

Intrusion Usage
Legitimate Usage of Non-Essential Functional Services
Legitimate Usage of Essential Functional Services

Figure 4: The Relationship Between Legitimate and Intrusion Usage

Development Requirements: Survivability places stringent requirements on system development and testing practices. Inadequate functionality and software errors can have a devastating effect on system survivability and provide opportunities for intruder exploitation. Sound engineering practices are required to create survivable software.

The following five principles (four technical and one organizational) are example requirements for survivable-system development and testing practices:

- Precisely specify the system's required functions in all possible circumstances of system use.
- Verify the correctness of system implementations with the system's functional specifications.
- Specify the use of system functions in all possible circumstances of system use, including intruder use.
- Test and certify the system based on function use and statistical methods.
- Establish permanent readiness teams for system monitoring, adaptation, and evolution.

Sound engineering practices are required to deal with legacy and COTS software components as well.

Operations Requirements: Survivability places demands on requirements for system operation and administration. These requirements include defining and communicating survivability policies, monitoring system use, responding to intrusions, and evolving system functions as needed to ensure survivability as usage environments and intrusion patterns change over time.

Evolution Requirements: System evolution responds to user requirements for new functions. However, this evolution is also necessary to respond to increasing intruder knowledge of system behavior and structure. In particular, survivability requires that system capabilities evolve more rapidly than intruder knowledge. This rapid evolution prevents intruders from accumulating information about otherwise invariant system behavior that they need to achieve successful penetration and exploitation.

2.1.1 Requirements Definition for Essential Services

The preceding discussion distinguishes between essential and non-essential services. Each system requirement must be examined to determine whether it corresponds to an essential service. The set of essential services must form a viable subsystem for users that is complete and coherent. If multiple levels of essential services are required, each set of services provided at each level must also be examined for completeness and coherence. In addition, requirements must be defined for making the transition to and from essential-service levels.

When distinguishing between essential and non-essential services, all of the usual requirements-definition processes and methods can be applied. Elicitation techniques such as those embodied in Software Requirements Engineering can help to identify essential services [Ebert 97]. Tradeoff and cost/benefit analysis can help to determine the sets of services that sufficiently address business survivability risks and vulnerabilities. Provisions for tracing survivability requirements through design, code, and test must be established. As previously mentioned, simulation of intrusion through intruder-use scenarios are included in the testing process.

2.1.2 Requirements Definition for Survivability Services

After specifying requirements for essential and non-essential services, a set of requirements for survivability services must be defined. These services can be organized into four general categories: resistance, recognition, recovery, and adaptation and evolution. These survivability services must operate in an intruder environment that can be characterized by three distinct phases of intrusion: penetration, exploration, and exploitation.

Penetration Phase. In this phase, an intruder attempts to gain access to a system through various attack scenarios. These scenarios range from random inputs by hobbyist hackers to well-planned attacks by professional intruders. These attempts are designed to capitalize on known system vulnerabilities.

Exploration Phase. In this phase, the system has been penetrated and the intruder is exploring internal system organization and capabilities. By exploring, the intruder learns how to exploit the access to achieve intrusion objectives.

Exploitation Phase. In this phase, the intruder has gained access to desired system facilities and is performing operations designed to compromise system capabilities.

Penetration, exploration, and exploitation create a spiral of increasing intruder authority and a widening circle of compromise. For example, penetration at the user level is typically a means to find root-level vulnerabilities. User-level authorization is then employed to exploit those vulnerabilities to achieve root-level penetration. Finally, compromise of the weakest host in a networked system allows that host to be used as a stepping-stone to compromise other more protected hosts.

Requirements definitions for resistance, recognition, recovery, and adaptation and evolution services help select survivability strategies to deal with these phases of intrusion. Some strategies, such as firewalls, are the product of extensive research and development and currently are used extensively in bounded networks. New survivability strategies are emerging to respond to the unique challenges of unbounded networks.

Resistance Service Requirements. Resistance is the capability of a system to deter attacks. Resistance is thus important in the penetration and exploration phases of an attack, before actual exploitation. Current strategies for deterring resistance include the use of firewalls, authentication, and encryption. Diversification is a resistance strategy that will likely become more important for unbounded networks.

Requirements for diversification must define planned variation in survivable system function, structure, organization, and the means for achieving it. Diversification is intended to create a *moving target* and render ineffective the accumulation of system knowledge as an intrusion strategy. Diversification also eliminates intrusion opportunities associated with multiple nodes that execute identical software and typically exhibit identical vulnerabilities. Such systems offer tempting economies of scale to intruders, since when one node has been penetrated, all nodes can be penetrated. Requirements for diversification can include variation in programs, retained data, and network routing and communication. For example, systematic means can be defined to randomize software programs while preserving functionality [Linger 98].

Recognition Service Requirements. Recognition is the capability of a system to recognize attacks or the probing that precedes attacks. Reacting or adapting during an intrusion is central to the capacity of a system to survive an attack that cannot be completely repelled. To react or adapt, the system must first recognize it is being attacked. In fact, recognition is essential in all three phases of attack.

Current strategies for attack recognition include both state-of-the-art intrusion detection and mundane but effective techniques such as logging applications and systems, administrative systems, frequent auditing, and follow-up investigations of reports generated by ordinary error detection. Advanced intrusion-detection techniques are generally of two types: anomaly detection and pattern recognition. *Anomaly detection* is based on models of normal user behavior. These models are often established through statistical analysis of system-use patterns. Deviations from normal system-use patterns are flagged as suspicious. *Pattern recognition* is based upon models of intruder behavior. User activity that matches a known pattern of intruder behavior raises an alarm.

Requirements for future survivable networks will likely employ additional strategies such as self-awareness, trust maintenance, and black-box reporting. Self-awareness is the process of establishing a high-level semantic model of the computations that a component or system is executing or has been asked to execute. A system or component that *understands* what it is being asked can refuse requests that would be dangerous, compromise a security policy, or adversely impact the delivery of minimum essential services.

Trust maintenance is achieved by a system through periodic queries among its components of (e.g., among the nodes in a network) to continually test and validate trust relationships. Detection of signs of intrusion would trigger an immediate test of trust relationships.

Black-box reporting is a dump of system information that can be retrieved from a crashed system or component for analysis to determine the cause of the crash (e.g., design error or specific intrusion type). This analysis can help to prevent other components from suffering the same fate.

A survivable-system design must include explicit requirements for recognition of attack. These requirements ensure the use of one or more of the preceding strategies through the specification of architectural features, automated tools, and manual processes. Since intruder techniques are constantly advancing, recognition requirements should be frequently reviewed and continuously improved.

Recovery Service Requirements. Recovery is a system's ability to restore services after an intrusion has occurred. Recovery also contributes to a system's ability to maintain essential services during intrusion.

Requirements for recoverability are what most clearly distinguish survivable systems from systems that are merely secure. Traditional computer security leads to the design of systems that rely almost entirely on hardening (i.e., resistance) for protection. Once security is breached, damage may follow with little to stand in the way. The ability of a system to react during an active intrusion is central to its capacity to survive an attack that cannot be completely repelled. Recovery is thus crucial during the exploration and exploitation phases of intrusion.

Recovery strategies in use today include replication of critical information and services, use of fault-tolerant designs, and incorporation of backup systems for hardware and software. These backup systems include master copies of critical software in isolation from the network. Some systems, such as large-scale transaction processing systems, employ elaborate, fine-grained transaction roll-back processes to maintain the consistency and integrity of state data.

Adaptation and Evolution Service Requirements. Adaptation and evolution are critical to maintaining resistance to ever-increasing intruder knowledge of how to exploit otherwise unchanging system functions. Dynamic adaptation permanently improves a system's ability to resist, recognize, and recover from intrusion attempts. For example, an adaptation requirement may be an infrastructure that enables the system to inoculate itself against newly-discovered security vulnerabilities by automatically distributing and applying security fixes to all network elements. Another adaptation requirement may be that intrusion detection rule sets are updated regularly in response to reports of known intruder activity from authoritative sources of security information, such as the CERT Coordination Center.

Adaptation requirements ensure that such capabilities are an integral part of a system's design. As in the cases of resistance, recognition, and recovery requirements, the constant evolution of intruder techniques requires that adaptation requirements be frequently reviewed and continuously improved.

3. Survivability Design and Implementation Strategies

In this section we examine strategies that support the survivability of critical system functions in unbounded networks. Strategies for survivability in networked systems depend on several assumptions and constraints. Although they may seem obvious, these assumptions and constraints must be made explicit. The assumptions differ radically from the implicit assumptions traditionally made for the uniprocessor, multiprocessor, and bounded network systems on which most previous research and development has been based.

For unbounded networks, we assume that

- any individual node of the network can be compromised
- survivability does not require that any particular physical component of the network be preserved
- only the essential services of the network as a whole must survive
- for reasons of reliability, design error, user error, and intentional compromise, the trustworthiness of a network node or any node with which it can communicate cannot be guaranteed

In this report, we primarily discuss unbounded networks. The term *unbounded* has a slightly different meaning depending on the purpose and situation involved. In all cases, unbounded networks relate to three principle characteristics that are present in each definition: a lack of central physical or administrative control, absence of insight or vision into all parts of the network, and no practical limit on growth in the number of nodes in the network.

These assumptions impose the following constraints on the architecture of survivable networks and on the form of feasible survivability strategies:

- There must not be a single point of failure within the network. Essential services are distributed in a manner that is not critically dependent on any particular component or node.
- Global knowledge is impossible to achieve in a distributed system [Halpern84]. There are no all-seeing global oracles. Instead, protocols define the interaction and knowledge shared between nodes.
- Each node must continuously validate the trustworthiness of itself and those with which it communicates.
- Computations within a given node of an bounded network, whether for essential services, communication, or trust validation, must have costs that are less than proportional to the number of nodes in the network.

3.1 Four Aspects of Survivability Solution Strategies

As introduced in Section 2, there are four aspects of the survivability solution which can serve as a basis for survivability strategies. These four aspects are: resistance, recognition, recovery, and system adaptation and evolution. This section summarizes the approaches in each of these four areas.

There are many techniques for dealing with these four aspects. Any or all of the techniques may apply to survivable systems. We do not list all of these techniques but instead categorize them within the broader aspects. Table 2 contains the four aspects of the survivability solution and representative taxonomies of respective strategies.

Survivability Aspect	**Taxonomies of Strategies**
Resistance	• traditional security, including encryption and covert channels • diversity and maximized differences in individual nodes • analytic redundancy and voting • specialization, division of labor, trust, and information • continuous validation of trust • exhibited stochastic properties and random behavior
Recognition	• analytic redundancy and testing (including failures in software, encryption, and trust) • intrusion monitoring and suspicious activities • system behavior and integrity monitoring
Recovery	• physical and information redundancy • non-local copies of information resources • preparation, readiness, contingency planning, and response teams
Adaptation and Evolution	• general or specific changes to resist, recognize, or recover from new vulnerabilities that are discovered • broadcast of warnings to other nodes • broadcast of adaptation and evolution strategies • deterrence through retaliation or punishment

Table 2: A Taxonomy of Strategies Related to Survivability

3.2 Support of Strategies by the Current Computing Infrastructure

The rapid growth of the Web and other Internet-based applications has encouraged the growth of a computing infrastructure to support distributed applications. While the initial Web efforts concentrated on information publishing, the application domain has expanded to encompass a much wider spectrum of an organization's computing needs. The technical focus of this growth has moved from tools such as Web browsers or servers to the development of a set of Internet-compatible, commercially provided services. Examples of these services are file, print, transaction, messaging, directory, security, and object services such as CORBA (Common Object Request Broker Architecture) and DCOM (Distributed Component Object Model).

The commercially available distributed infrastructures are in the early phases of their development and do not yet directly support system survivability. Recognition is not a supported service and recovery is indirectly supported by a transaction server. Typically, an organization adopts such an infrastructure to lower costs by using a common infrastructure for intranets, extranets, and Internet applications and to simplify application development by capturing the complexity of distributed computing in the infrastructure rather than in each application.

Managing user-profile data is an example of a service that a distributed infrastructure can assume. One general requirement of system survivability is to provide user authentication and manage the authority given to that user for data and systems access. Authentication can be implemented using passwords and authorizations that are validated by access-control lists. However, in many existing systems, such as database applications, access-control lists are maintained by the application.

When system users, data, and applications are geographically distributed, the maintenance of user-profile data in an application is difficult. A shared directory service, which is part of a distributed infrastructure, can provide the data storage capability and a protocol such as LDAP (Lightweight Directory Access Protocol) for application access and replace the application-specific access-control mechanisms. These infrastructure security services can provide the mechanisms for user authentication such as a public key interface, mechanisms to describe access control, and the means to define a security policy. The use of shared services for user authentication and authorization should reduce application and overall system complexity as well as provide the means to define an organizational security policy.

When this strategy is implemented, the system architecture is constrained by the infrastructure-supplied services and the protocols supported. For example, a survivability strategy may be to exchange a primary service with an alternate implementation of that service if the primary service has been compromised. At this stage of infrastructure deployment there is some interoperability supported among services provided by different vendors, however, there is also significant integration of services that makes it difficult or impossible to replace a service, such as a directory service, with one from a different vendor.

Using shared directory services also raises general survivability issues. A widely used infrastructure should develop a robust set of services. However, their wide use develops a large and knowledgeable intruder community and a wide dissemination of information about system vulnerabilities and security solutions. A compromised or inaccessible directory can affect multiple applications and multiple sites.

An essential part of providing system survivability is establishing operational and administrative procedures for system directories so that system administrators can monitor service and provide recovery. The design tradeoff is that implementing monitoring and recovery procedures is less costly using shared components than using an application-specific architecture. Infrastructure services provide generic support for replication and maintenance of consistency across distributed sites. However, achieving overall mission survivability requires not only understanding the impact of compromised access control data and of the design of a recovery policy, but also knowledge of the system's applications.

Commercially available infrastructure products provide general services that are independent of application domain. Some of the services listed in Figure 3, however, require application-domain knowledge. For example, recognition of an intrusion or maintenance of trust among nodes requires knowledge of expected behavior. A protocol can ensure that information is delivered, but cannot validate the appropriateness of the data. Simple recovery mechanisms can include transaction logs or file restorations; but use of transactions, rollback strategies, and more advanced techniques require domain expertise to identify consistent application states and the impact of compromised data. The successful use of such recovery strategies has been in application-centered products, such as relational database systems that manage relatively homogeneous data structures. Applying such techniques to general distributed-computing systems is more difficult.

3.3 Survivability Design Observations

We can draw a number of observations about the questions and issues that must be addressed concerning system survivability in networked systems.

3.3.1 Survivability Requires Trust Maintenance

An open issue is how to determine the basis of trust and how an individual node of a network contributes to the survivability of the system's essential services when

- any node can be unreliable or rogue
- there is no global view or global control
- nodes cannot completely trust themselves or their neighbors

Depending on the application, it may be possible through architectural design or dynamic action within the system to increase the reliability, visibility, and control of components or the trustworthiness of participants. The only absolute basis for trust maintenance, however, is the consistency of behavioral feedback from interactions with other nodes and independent verification of claimed actions from nodes not directly involved in the transactions.

A closely related point is the absence of global view and control. If unreliable and untrustworthy components are found to be present in a system, determining whether the critical functions have been compromised may be extremely difficult without global view and control. If global view and control are absent (and, in general, they will be) this condition does not preclude effective survivable-network architectures. In particular, it should be possible for individual nodes to generally contribute to the survivability goals and at worst not interfere with these goals.

Genetic algorithms, for example, achieve their effects through the collective action of the individual participants. These participants, however, cannot measure overall effectiveness or determine whether their contribution is positive. This example suggests that survivability solutions can exist among emergent algorithms that depend on continuous interaction with neighboring nodes but do not require feedback for indications of progress and success.

3.3.2 Survivability Analysis Is Protocol-Based Not Topology-Based

Another implication for networked systems is that the important aspects of their architecture from the viewpoint of survivability relate to the conventions and rules of interaction between neighboring nodes and that the network topology is largely irrelevant. That is, network architectures must be specified, compared, and measured in terms of their interactions and not the topology of their interconnection.

As an example of this kind of analysis, consider the general issue of persistence of state data for a protocol. Should a protocol maintain state information to improve reliability or to perform additional consistency checks? What level of checking should the infrastructure support? J. H. Saltzer and his colleagues examined the FTP (file transfer protocol) and compared approaches that check packets only at the source and destination nodes (end-to-end) to protocols that check reliability on each hop of the communications path [Saltzer 84]. The conclusion was that hop-to-hop checking increased complexity and affected performance with little increase in overall reliability.

Kenneth P. Birman discusses such tradeoffs in a more general context [Birman96]. Properties such as reliability and survivability can be enhanced by properties that support fault tolerance or communication guarantees. However, the cost of a property to support, say uniform ordering of events, can be thousands of times more costly than a weaker property that may require the application to handle nonuniform behavior.

Similar arguments can be made when you compare stateless architectures and non-replicated data to maintaining a strong application-level consistency requirement. In the case of stateless architectures and non-replicated data, the server can be restarted and the clients have the responsibility to reconnect. Survivability requires tradeoff analysis between the responsibilities of the servers and the clients and between end-to-end protocol monitoring by the application and general protocol monitoring provided by the infrastructure. For such a recovery strategy, the application level may be the appropriate level in which to analyze application-state and user behavior and select appropriate recovery actions.

3.3.3 Survivability Is Emergent and Stochastic

Survivability goals are emergent properties that are desired for the system as a whole, but do not necessarily prevail for individual nodes of the system. This approach contrasts with traditional system designs in which specialized functions or properties are assured for particular nodes and the composition of the system must ensure that those properties and functional capabilities are preserved for the system as a whole. For survivability, we must achieve system-wide properties that typically do not exist in individual nodes. A survivable system must ensure that desired survivability properties emerge from the interactions among the components in the construction of reliable systems from unreliable components.

Survivability is inherently stochastic. If survivability properties are emergent, they are present only when the number of contributing component nodes of a system is sufficiently large. If the number or arrangement of nodes falls below a critical threshold, the attendant survivability property fails. An example of this type of critical survivability property is connectivity in a communications system.

You can design the architecture of the system to maximize the number of paths between any two nodes; but if enough links are compromised to partition the network, communication between arbitrary nodes will no longer succeed. Thus, survivability properties, algorithms, and architectures should be specified, viewed, and assessed to determine the probability of their success under given conditions of use and not determined as discrete quantities.

3.3.4 Survivability Requires a Management Component

The design of a survivable system also includes management operations and administration. Poor system administration is a frequent source of vulnerabilities at centrally administered sites. In unbounded network systems, system administration must be coordinated across multiple sites. Existing system administration procedures typically assume a bounded environment and full administrative control over the required services. The complexity of infrastructure and the use of services outside an organization's immediate control require expanding the administrative services and providing a monitoring function as part of the infrastructure.

4. A Survivability Engineering Process

One approach in the analysis and synthesis of survivable systems is to consider survivability as a composite property that consists of many quality attributes balanced to enhance the overall survivability of the system. In large systems, system quality depends as much on software architecture as on code-level practices such as programming language, detailed design, algorithms, data structures, and testing. In this section, we explore architecture issues and propose a process that enables an architecture to support survivability as a system attribute.

4.1 Architecture-Based Development of Survivable Systems

An attribute such as survivability does not exist in isolation. A system has multiple quality attributes such as performance, availability, and modifiability. Attributes and their analyses interact. Performance affects modifiability. Availability affects safety. Security affects performance. Everything affects cost. While experienced system designers know that these tradeoffs exist, no codified method exists for characterizing quality attributes and, in particular, characterizing their interactions.

Often, system designers neglect to consider survivability and security in their designs. Typically, survivability is considered in isolation from other attributes of software-engineering quality, such as performance, dependability, modifiability, and ease of use. This approach is not surprising, since in traditional software engineering the other quality attributes dominate the design process whereas, survivability and security (if considered at all) are usually an afterthought.

Software designers, their managers, and their customers must be able to specify the tradeoffs between enhanced survivability and other attributes of software quality (including affordability). These groups also must be able to evaluate how well competing designs (and implementations) achieve overall system specifications, including survivability specifications.

4.1.1 Survivability as an Add-On Patch

Survivability and security typically are relegated to a series of add-on patches that are put in place after problems are discovered and reported either by the customer or by an incident response team. Because these patches are typically quick reactions to an emergency situation, rather than the result of principled systems engineering design, they do not solve broad classes of problems. These patches often solve only a small number of problems and leave many others unsolved. In fact, they occasionally introduce new security vulnerabilities (or bring old ones back to life).

When a patch is designed, the development focus is often on the speed and ease of the solution, and on maintaining performance and functionality (e.g., don't degrade existing abilities) rather than on designing the best solution from a survivability or security standpoint. Conversely, those designing a patch sometimes decide to maximize security and survivability at any cost. Using either approach, leads to design efforts in which tradeoffs are not appropriately considered.

Changes to software systems are rarely made at the architectural level. For example, the patches for security vulnerabilities are not usually reflected in the architectural description of the system. This disconnect between the architecture and the implementation of a system frequently has an unexpected and undesirable impact on security and other attributes of survivability. Daily, the CERT Coordination Center sees the real-world damage (system intrusions) caused by the lack of a theoretical foundation upon which to build sound software engineering practices for the design, implementation, and maintenance of survivable systems.

The characteristics and dimensions of the design tradeoffs that arise once survivability is made an inherent part of the software-engineering process is a fertile area for exploratory and applied research. This area has the potential for a high payoff in software engineering process improvement.

4.1.2 A Scenario-Based Architecture Design Process

Survivability issues affect system requirements in several areas. In Section 2, we identified four requirements categories of system and survivability functions:

- usage, including intrusion use
- development methodology
- system operation
- system evolution

In this section, we concentrate on an approach that addresses the first two requirement categories, specifically, how to evaluate the capability of a system to deliver essential functions in an environment that includes intrusion scenarios. Our general approach to survivability and security is consistent with the SEI-developed Architecture Tradeoff Analysis (ATA) [Kazman 97].

The ATA approach is based on

- a set of system attributes
- analytic measures of the system that are based upon formal models (e.g., performance and availability)
- qualitative measures of the system that are based upon formal inspections (e.g., modifiability, safety, and security)

Each of these measures evaluates the software architecture along a distinct dimension. Taken together, these dimensions are of interest to the system's stakeholders. Figure 5 illustrates the ATA process. The scenario-based Software Architecture Analysis Method (SAAM) applied to modifiability and extensibility is described in [Kazman 96].

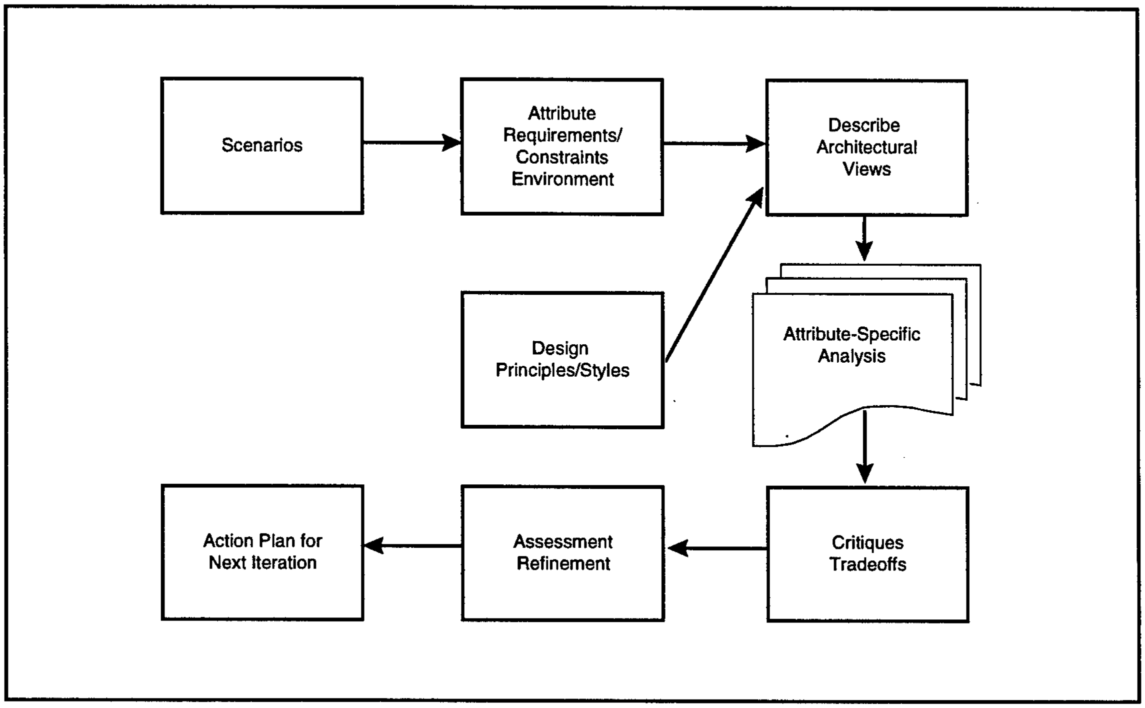

Scenarios
Attribute Requirements/ Constraints Environment
Describe Architectural Views
Design Principles/Styles
Attribute-Specific Analysis
Action Plan for Next Iteration
Assessment Refinement
Critiques Tradeoffs

Figure 5: Attribute Tradeoff Analysis Process

Scenarios are a way to evaluate a proposed architecture before the system is built. For intrusion requirements, this evaluation involves generating possible attack scenarios and evaluating the capability of the system to resist, recognize, and recover from such attacks, and to adapt and evolve so as to limit the effectiveness of future attacks.

A scenario is not necessarily an explicit script for breaking into a system. Scenarios can focus on the impact of having a critical system component, such as user authentication, compromised and on how to recover once that situation occurs. Generating such scenarios requires looking at the system from the intruder's perspective. Intruder's often have limited resources and their strategies often depend on

- exploiting known weak spots in technology
- identifying system dependencies and weak links
- monitoring network communication between components to capture information, monitor activity, and disrupt communications

Scenarios should identify system hot spots that provide opportunities for successful intrusions. This hot-spot method is a practical approach ideally suited for contingency planning to combat an intelligent adversary. Contingency planning includes analyzing survivability for as many intruder scenarios as practical. Scenarios also document those aspects of a bounded environment that the system is designed to confront.

The scenarios should account for intrusion objective, impact, strategies, and properties.

4.1.2.1 Intrusion Objective

The intent of intrusion can include denial of service, access to confidential data, or compromise of existing data.

4.1.2.2 Intrusion Impact

The impact of intrusion on a system can be direct or indirect. Direct impact can affect performance or availability. Indirect impact can result in loss of business and customer trust. Data collected on intrusion impact is part of the cost-benefit analysis used in the tradeoff analysis of system properties.

4.1.2.3 Intrusion Strategies

An intrusion strategy is a technique that helps intruders achieve one or more intrusion objectives. The set of intrusion strategies is unlimited. The combination of the strategies that an intruder chooses reflects the specific objectives and knowledge of that intruder. Intrusion strategies include established techniques and new approaches.

4.1.2.4 Intrusion Properties

Intrusion properties are quantitative properties that systems exhibit when they are successfully penetrated and compromised. Each property can be defined by the observable system effects it produces. The set of properties is unlimited. The combination of properties that an intruder chooses for given set of intrusion requirements reflects the specific objectives of that intruder. Some properties, such as root privileges, may be achieved by exploiting system vulnerabilities, or by other means, such as compromising a system administrator.

The ATA process evaluates an architecture using a collection of system properties. The tradeoff analysis regarding security and survivability consists of several tasks:

- Generate intrusion scenarios.
- Establish priorities and costs associated with the effects of the scenarios.
- Identify architecture hot-spots based on effects of the intrusion scenarios.
- Generate and evaluate security and survivability strategies for each proposed architecture regarding the hot spots.
- Identify the requirements for system-level services such as user authentication and authorization that may be necessary for security or survivability.

Some hot spots will be associated with known vulnerabilities of the proposed technologies. A strategy to address this type of situation may be an architecture that supports an alternate implementation of a service (e.g., Ethernet and Token Ring for a local area network) or permits the rapid upgrade of a service, such as a Web server, to fix a new vulnerability.

Hot spots may be associated with the general communications topology of the system. For example, a central database or directory service may maintain user profile and authentication information. There may be specific strategies to protect that information and restrict system access if there are communications problems.

Hot spots may also be associated with application logic and the protocols used to exchange information among distributed components. As we discussed in Section 3, there is limited global knowledge that is available about this subject. Analysis should concentrate on pair-wise exchanges of data. Components that maintain extensive state information may complicate recovery.

4.2 An Architecture-Based Survivability Software Process

Figure 6 summarizes an architectural evaluation process for survivability based on the principles of architecture evaluation through system-use scenarios. This evaluation process regards intruders as another class of user, on par with the legitimate users of a system. This process is an extension and specialization of the ATA process described in Figure 5.

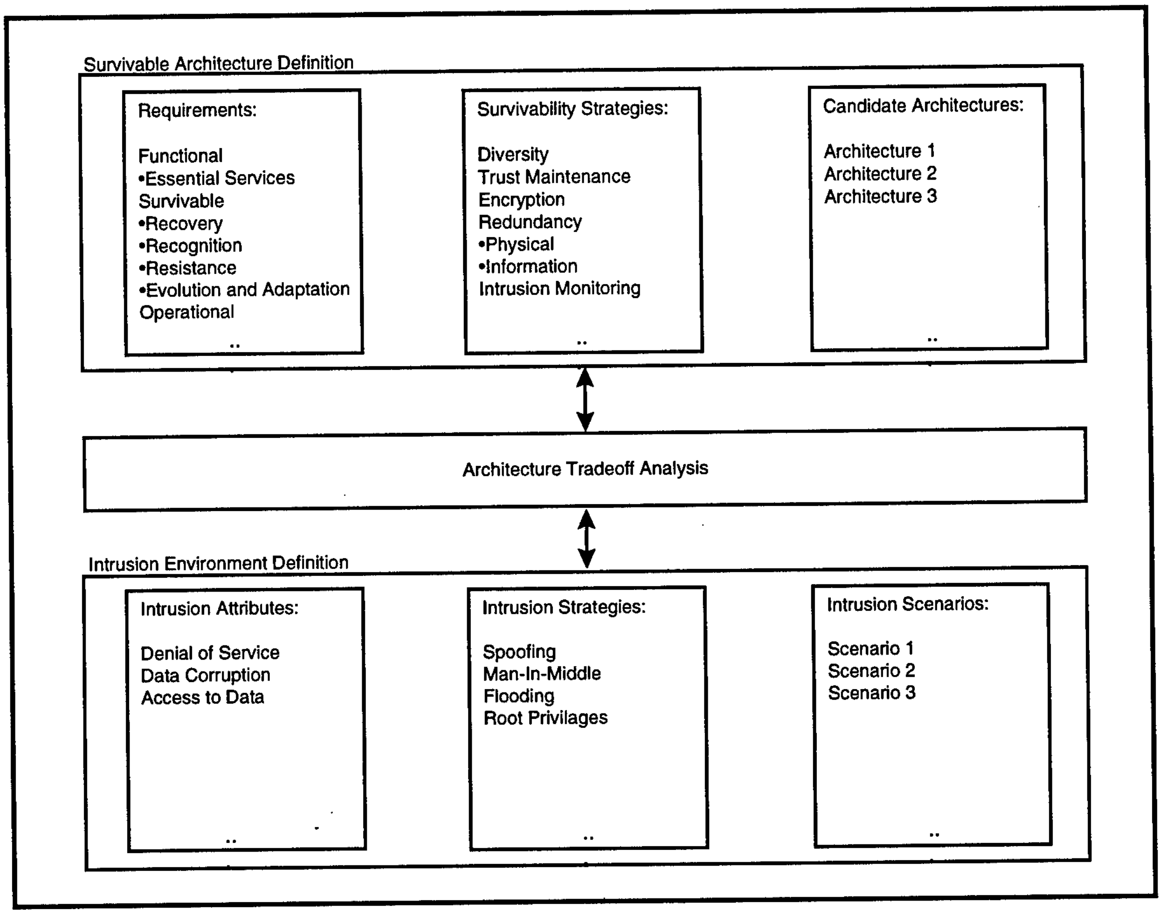


Figure 6: Survivable Architecture Tradeoff Analysis Process

The evaluation process permits analysis of the interaction between survivable architectures and intrusion environments through three major activities: survivable architecture definition, intrusion environment definition, and survivable architecture tradeoff analysis.

4.2.1.1 Survivable Architecture Definition

Survivability functions can be defined and embedded within candidate architectures based on survivability requirements and strategies. These architectures also include system functions that are required by users.

4.2.1.2 Intrusion Environment Definition

Intrusion capabilities can be defined and embedded within system-use models based on intrusion requirements and strategies. These system-use models also include the uses of system functions that are required by users.

4.2.1.3 Survivable Architecture Tradeoff Analysis

The performance of candidate architectures can be analyzed using scenarios that are generated by the system-use models. This analysis results in feedback to requirements and possible modifications to architecture and system-use definitions for subsequent analysis.

We plan to evaluate and report on the effectiveness of this evaluation process by applying it in several pilot studies.

5. Research Directions

There are a number of promising research areas in survivable systems. The plans for the Survivable Network Technology team at the SEI include

- adapting and developing architectural description techniques to adequately describe large-scale distributed systems with survivability attributes
- representing intruder environments through intruder usage models
- creating an analysis method to evaluate survivability as a global emergent property from architectural specification
- refining the analysis technology and instruments through pilot tests of real distributed systems

Bibliography

[Anderson 97] Anderson, R. H.; Hearn, A. C.; & Hundley, R. O. *RAND Studies of Cyberspace Security Issues and the Concept of a U.S. Minimum Essential Information Infrastructure* [online]. Available WWW <URL: http://www.cert.org/research/isw97_hypertext/all_the_papers/no1.html> (1997).

[Bass 98] Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice.* Reading, Mass.: Addison Wesley Longman, 1998.

[Birman 96] Birman, Kenneth P. *Building Secure and Reliable Network Application.* Greenwich, Connecticut: Manning, 1996.

[Clark 93] Clark, R. K.; Greenberg, I. B.; Boucher, P. K.; Lund, T. F.; Neumann, P. G.; Wells, D. M.; & Jenson, E. D. "Effects of Multilevel Security on Real-Time Applications," 120-129. *Proceedings of Ninth Annual Computer Security Applications.* Orlando, Florida, December 6-10, 1993. Los Alamitos, Ca.: IEEE Computer Society Press, 1993.

[Ebert 97] Ebert, C. "Dealing with Nonfunctional Requirements in Large Software Systems." Annals of Software Engineering 3 (September 1997): 367-395.

[Halpern 84] Halpern, J. & Moses, Y. "Knowledge and Common Knowledge in a Distributed Environment," 50-61. *Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing.* Vancouver, British Columbia, Canada, August 27, 1984. New York, N.Y.: Association for Computing Engineers (ACM), 1984.

[Kazman 96] Kazman, R.; Abowd, G.; Bass, L.; & Clements, P. "Scenario-Based Analysis of Software Architecture." *IEEE Software* 13, 6 (November 1996): 47-55.

[Kazman 97] Kazman, R.; Klein, M.; Barbacci, M.; Longstaff, T.; Lipson, H.; & Carriere, J. *The Architecture Tradeoff Analysis Method* [online]. Available WWW <URL: http://www.sei.cmu.edu/technology/product_line_systems/ata_method.html> (1997).

[Leveson 95] Leveson, N. G. *Safeware: System Safety and Computers*, New York, New York: Addison-Wesley, 1995.

[Linger 97] Linger, R.; Mead, N.; Lipson, H. *Requirements Definition for Survivable Network Systems* [online]. Available WWW < URL: http://www.cert.org/research> (1997).

[Linger 98] Linger, R. *Systematic Generation of Stochastic Diversity in Survivable System Software* [online]. Available WWW <URL: http://www.cert.org/research> (1998)

[Lipson 97] Lipson, H. & Longstaff, T. *Information Survivability Workshop* [online]. Available WWW <URL: http://www.cert.org/research/isw97_hypertext/front_page.html> (1997).

[Mendiratta 96] Mendiratta, V. "Assessing the Reliability Impacts of Software Fault-Tolerance Mechanisms," 99-103. *Proceedings of the Seventh International Symposium on Software Reliability Engineering.* White Plains, N.Y., October 30 to November 2, 1996. Los Alamitos, Ca.: IEEE Computer Society Press, 1996.

[Mills 92] Mills, H. D. "Certifying the Correctness of Software," vol 2, 373-381. *Proceedings of 25th Hawaii International Conference on System Sciences.* Kauai, Hawaii, January 7-10, 1992. Los Alamitos, Ca.: IEEE Computer Society Press, 1992.

[Musa 87] Musa, J.; Iannino, A.; & Okumoto, K. *Software Reliability: Measurement, Prediction, and Application.* New York, N.Y.: McGraw-Hill, 1987.

[Saltzer 84] Saltzer, J. H.; Reed, D. P.; & Clark, D. D. "End-to-End Arguments in System Design." *ACM Transactions on Computer Systems* 2, 4 (November 1984): 277-288.

[Trammell 95] Trammell, C. "Quantifying the Reliability of Software: Statistical Testing Based on a Usage Model," 208-218. *Proceedings of the Second IEEE International Symposium on Software Engineering Standards.* Montreal, Quebec, Canada, August 21-25, 1995. Los Alamitos, Ca.: IEEE Computer Society Press, 1995.

Glossary

Adaptation and Evolution Services	Survivable system functions provided to continually improve the system's capability to deliver essential services, typically by improving resistance, recognition, and recovery capabilities
Essential Services	Services to users of a system that must be provided even in the presence of intrusion, failure, or accident
Intrusion	An attack on a network for purposes of gaining access to or destroying privileged information, or disrupting services to legitimate users
Network Architecture	A definition of the high-level behavior of and connections among nodes in a network, sufficient to evaluate network properties
Non-Essential Services	Services to users of a system that can be temporarily suspended to permit delivery of essential services while the system is dealing with intrusions and compromises.
Recognition Services	Survivable system functions that detect attempted and successful intrusions
Recovery Services	Survivable system functions that restore full services after an intrusion has occurred
Resistance Services	Survivable system properties and functions that make intrusion difficult and costly
Survivability	The capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents
Survivability Requirements	The definition of essential services as well as resistance, recognition, recovery, and adaptation and evolution functions that are sufficient to achieve required levels of a system's survivability
System Requirements	The definition of user requirements for system services and usage, for which survivability requirements can be defined
Unbounded Network	A network characterized by topology and functionality that cannot be determined, and by the absence of centralized administrative control

Usage Model A definition of all possible usage scenarios of a system, including legitimate and intruder use

Usage Scenario An instance of system use, either legitimate or intruder use

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. **AGENCY USE ONLY** (LEAVE BLANK)

2. **REPORT DATE**
November 1997

3. **REPORT TYPE AND DATES COVERED**
Final

4. **TITLE AND SUBTITLE**
Survivable Network Systems: An Emerging Discipline

5. **FUNDING NUMBERS**
C — F19628-95-C-0003

6. **AUTHOR(S)**
R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. Longstaff, & N. R. Mead

7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. **PERFORMING ORGANIZATION REPORT NUMBER**
CMU/SEI-97-TR-013

9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. **SPONSORING/MONITORING AGENCY REPORT NUMBER**
ESC-TR-97-013

11. **SUPPLEMENTARY NOTES**

12.A **DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified/Unlimited, DTIC, NTIS

12.B **DISTRIBUTION CODE**

13. **ABSTRACT** (MAXIMUM 200 WORDS)

Society is growing increasingly dependent upon large-scale, highly distributed systems that operate in unbounded network environments. Unbounded networks, such as the Internet, have no central administrative control and no unified security policy. Furthermore, the number and nature of the nodes connected to such networks cannot be fully known. Despite the best efforts of security practitioners, no amount of system hardening can assure that a system that is connected to an unbounded network will be invulnerable to attack. The discipline of survivability can help ensure that such systems can deliver essential services and maintain essential properties such as integrity, confidentiality, and performance, despite the presence of intrusions. Unlike the traditional security measures that require central control or administration, survivability is intended to address unbounded network environments. This report describes the survivability approach to helping assure that a system that must operate in an unbounded network is robust in the presence of attack and will survive attacks that result in successful intrusions. Included are discussions of survivability as an integrated engineering framework, the current state of survivability practice, the specification of survivability requirements, strategies for achieving survivability, and techniques and processes for analyzing survivability.

14. **SUBJECT TERMS:** survivability, security,unbounded networks, networks, Internet

15. **NUMBER OF PAGES**
46

16. **PRICE CODE**

17. **SECURITY CLASSIFICATION OF REPORT**
UNCLASSIFIED

18. **SECURITY CLASSIFICATION OF THIS PAGE**
UNCLASSIFIED

19. **SECURITY CLASSIFICATION OF ABSTRACT**
UNCLASSIFIED

20. **LIMITATION OF ABSTRACT**
UL